Helsinki University of Technology
CFD-group/ Laboratory of Applied Thermodynamics

MEMO No CFD/TERMO-32-2000 DATE: April 30, 2000

TITLE

SIMULATION OF A FREE-SURFACE FLOW

AUTHOR(S)
Patrik Rautaheimo and Esa Salminen

ABSTRACT

A numerical model has been developed for a simulation of moving boundaries. The flow field is calculated in one side of the moving interface and pressure is used to determine the boundary condition. The method is based on the moving-grid technique. The flow field is solved by an existing computational fluid dynamics code that uses a finite volume approach. The code uses structured multiblock grids, artificial compressibility method, higher-order discretization and implicit time marching.

MAIN RESULT

Numerical study of a flow around ships

PAGES
23

KEY WORDS
free-surface flow, finite-volume method, artificial compressibility.

APPROVED BY

Timo Siikonen April 30, 2000

1 Introduction

Flows with moving interfaces are of great interest in technical processes. These flows can be found in numerous cases: mold filling, flows past ships, and free jets are just a few examples of this type of the problems. In many flow cases the shape of the surface or interface is important for the whole behaviour of the physical system. The interface shape can be very complex and thus difficult to determine numerically.

In this study a surface tracking method is used. The grid is generated after every iteration cycle. This is done by using the original grid and by moving grid points along the grid lines in one direction. Basically, grid lines could be even moved into multidirections with this method. Limitation of this approach is that it cannot simulate complex interface shapes, like braking of waves. An advantage of the method is the accuracy of the flow field next to the free surface, including intersection of solid and free surfaces. In the previous studies with the present code, the moving grid method is implemented in a more complex manner [1]. In that approach grid points are allowed to move in more than one direction and a differential smoother is used to make a new grid. This procedure contains difficulties to assure a grid positivity. In the present novel approach the surface tracking equations are redesigned in order to keep the shape of the surface continuous, to enhance convergence and to improve the accuracy. The boundary conditions for the flow solver are given as fluxes at the free surface. Methods to take account surface tension are introduced. The surface tension plays an important role in getting the boundary layer surface height correctly and robustly.

Extra difficulties arise from the use of the artificial compressibility method. For the artificial compressibility method one must calculate a density change in a control volume that changes the pressure. It seems that the pressure field has a poor convergence rate with the present free-surface model. In this work, the mass fluxes are calculated by using a pressure-weighted interpolation of the surface velocity. This approach is somewhat similar to the pressure correction method.

The surface tracking method is built into a finite volume multi-block Navier–Stokes solver [2, 3]. The code has been applied for various external [4, 2, 5] and internal [6, 7, 8, 9] flows and thus, the validation of the viscous flow solver is not needed here. Instead the emphasis is put on the free surface boundary conditions. Turbulence can be modeled by an algebraic model, a two-equation model or by a Reynolds-stress model. In the present case, the algebraic model is used because of a fast convergence rate. Even though the code is parallelized [3, 10, 11], the free surface part is currently only working in a singe processor mode. In the following the main features of the flow solver are shortly reviewed, a method to track the free surface is described in detail, two test computations are presented, some future suggestion is made and conclusions are stated.

2 Methods

2.1 Basic Features of the Flow Solver

The flow simulation is based on the solution of the Reynolds averaged Navier–Stokes equations:

$$\frac{\partial U}{\partial t} + \frac{\partial (F - F_v)}{\partial x} + \frac{\partial (G - G_v)}{\partial y} + \frac{\partial (H - H_v)}{\partial z} = Q \tag{1}$$

where U is the vector of dependent variables, F, G, H and F_v, G_v, H_v represent the inviscid and viscous parts of the fluxes, and Q is a possible source term. The flow solver utilizes a structured multiblock grid. For the solution Eq. (1) is written in a finite-volume form

$$V_i \frac{dU_i}{dt} = \sum_{\text{faces}} -S(\hat{F} - \hat{F}_v) + V_i Q_i \tag{2}$$

where V_i is a cell volume, $\hat{F}$ and $\hat{F}_v$ are the inviscid and viscous parts of the flux on the cell surface. The sum is taken over the faces of the computational cell.

The solution proceeds blockwise after explicitly defined boundary conditions. The boundary conditions between the blocks are defined only on the highest grid level. In each block an implicit LU-factored solution with a multigrid acceleration of convergence is performed [12, 13]. The underlying solution method is based either on a flux-difference [14] or a flux-vector [15] splitting. The flux calculation utilizes a MUSCL-type differencing with a second- or a third-order accuracy. In this study a new type of a flux-difference splitting is introduced and third-order accurate upwind-biased scheme is used.

2.2 Free surface boundary conditions

Simulation of the wave height

The surface height is calculated by following the movement of the surface

$$\frac{\partial h}{\partial t} = \hat{w} \tag{3}$$

where $\hat{w}$ is a local normal velocity of the surface. Velocity is taken from the flow solver solution. It was found out that a region next to the solid surfaces has a slow convergence. In order to get the wave shape next to the solid surface, an artificial surface tension is added to the equation of the surface. Using an explicit Euler method with the surface tension term, the following form is obtained

$$\frac{\gamma}{\rho_\infty g}\left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}\right) + h = h_{old} + \Delta t \hat{w} \tag{4}$$

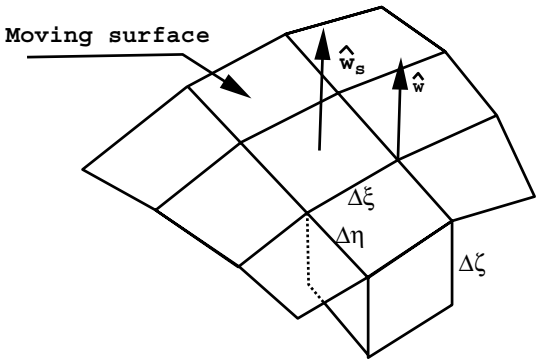

Fig. 1: Part of the grid next the free surface.

Currently γ is a model parameter with a value of 0.1N/m. The time-step size is calculated from the following equation obtained by a trial and error

$$\Delta t = CFL \min\left[\frac{1}{2}\min\left(\Delta\xi, \Delta\eta\right), 10^{-2} L_{ref}\right] / \left(V_{ref}\right) \tag{5}$$

where $\Delta\xi$ and $\Delta\eta$ are cell sizes parallel to the surface (see Fig. 1). However, on the right-hand side the term $\Delta t\hat{w}$, i.e. the maximum change of the height is also limited with "depth" of the first cell next to the surface

$$\left(\Delta t\hat{w}\right)_{mod} = \Delta t\hat{w}\frac{1}{8}\frac{\Delta\zeta}{|\Delta t\hat{w}| + \frac{1}{8}\Delta\zeta} \tag{6}$$

where $\Delta\zeta$ is the size of the grid perpendicular to the surface. Figure 1 shows the variables used. These parameters are not final and more tests and refinements should be done. The elliptic system of Eq. (4) is solved by line Gauss-Seidel iteration with only one iteration sweep. The first guess is the current height. The solved height h is relaxed by

$$h_{new} = 0.5h + 0.5h_{old} \tag{7}$$

The equations for the surface height should be written in a delta form in a future.

Boundary condition for the flow solver

In Navier–Stokes equations the boundary conditions can classified to an 'inlet type' or an 'outlet type.' In the inlet type, velocities are generally given and the pressure is extrapolated from the computational domain. For the outlet type, the pressure is set and a zero gradient is assumed for the velocities. In this case, the physical boundary conditions are zero pressure (or free-stream pressure) and zero velocity-gradients. However, this cannot be given directly for the solver, since the problem would be over-specified. The problem is solved so that the pressure is given and the surface of the grid is moved as long as the velocities are equal to zero at the free surface.

The boundary conditions at the free surface are given in the following way: For the velocities, the extrapolation from the flow field is done. However, the averaged velocity through the surface is calculated and it is used to correct velocities so that the total mass flux through the surface is zero in every iteration sweep. The pressure is given by using the height of the surface and also the curvature. Pressure is defined in the flow solver as

$$p = p_{dif} + p_\infty + \rho_\infty g h \tag{8}$$

where p_{dif} has a value close to zero everywhere in the flow field. The solved variable is p_{dif} and thus the boundary conditions must be given for it also. Here, a zero hydrostatic pressure is assumed as $h = 0$. Surface pressure $p_{dif,s}$ is calculated as

$$p_{dif,s} = -\rho_\infty g h - \mathcal{T}\left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}\right) \tag{9}$$

where $\mathcal{T}$ is the surface tension coefficient for air-water interface at temperature of 20^oC and have value of 0.0727N/m. The surface tension equation has not exactly the form of Eq. (9) [16] but

$$p_{dif,f} = -\rho_\infty g h - \mathcal{T}\left(\frac{1}{R_x} + \frac{1}{R_y}\right) \tag{10}$$

where R_x and R_y are the radii of curvature of the surface.

2.3 Numerical Issues

Simulation of the wave height

The surface height is calculated at the grid points whereas the flow solver information comes from the centers of the cells. The surface normal velocity $\hat{w}$ in Eq. (4) can be extrapolated from the cell center points with first-order or second-order methods. In the second-order method a zero gradient is assumed at the surface and following difference equation is obtained

$$\hat{w}_s = \frac{9\hat{w}_1 - \hat{w}_2}{8} \tag{11}$$

where $\hat{w}_1$ is the first computational point next to the solid surface and $\hat{w}_2$ is the second one. In the first-order method the surface velocity is taken as a value of the first computational point.

$$\hat{w}_s = \hat{w}_1 \tag{12}$$

After test computations the former was found to be marginally more stable and the second-order method did not give any improvement for the solved surface. Now $\hat{w}_s$ is defined at the center of the cell face. These values are needed at the corners of the cell faces. Values are taken from the upwind direction. This was found to be one of the essential features of the present procedure. An

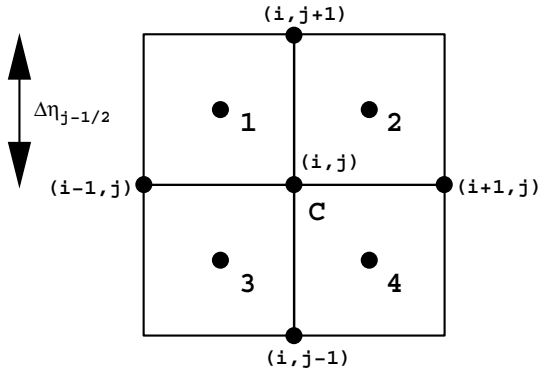

Fig. 2: Grid used to upwind the surface movement.

example of the grid is seen in Fig. 2. Values from points 1 and 4, or 3 and 2 are transferred to the grid point C as

$$\hat{w}_{14} = \alpha_{uw}\hat{w}_1 + (1-\alpha_{uw})\hat{w}_4, \text{ if flow is from 1 to 4} \quad (13)$$

$$\hat{w}_{14} = \alpha_{uw}\hat{w}_4 + (1-\alpha_{uw})\hat{w}_1, \text{ if flow is from 4 to 1} \quad (14)$$

$$\hat{w}_{23} = \alpha_{uw}\hat{w}_2 + (1-\alpha_{uw})\hat{w}_3, \text{ if flow is from 2 to 3} \quad (15)$$

$$\hat{w}_{23} = \alpha_{uw}\hat{w}_3 + (1-\alpha_{uw})\hat{w}_2, \text{ if flow is from 3 to 2} \quad (16)$$

where α_{uw} is an upwinding parameter and the value of $\alpha_{uw} = 0.75$ was found to perform well. A full upwinding or central differencing were not as stable. In order to get a combined value from $\hat{w}_{14}$ and $\hat{w}_{23}$, the following formula was used

$$\hat{w}_C = \frac{|\vec{n}_{14}\cdot\vec{u}_{14}|}{|\vec{n}_{14}\cdot\vec{u}_{14}| + |\vec{n}_{23}\cdot\vec{u}_{23}|}\hat{w}_{14} + \left(1 - \frac{|\vec{n}_{14}\cdot\vec{u}_{14}|}{|\vec{n}_{14}\cdot\vec{u}_{14}| + |\vec{n}_{23}\cdot\vec{u}_{23}|}\right)\hat{w}_{23} \quad (17)$$

where $\vec{n}_{ij}$ and $\vec{u}_{ij}$ are a normal vector parallel from point i to j, and an averaged velocity of the points i and j, respectively.

In Eq. (4), the surface height is solved implicitly. In a difference format the equation gets the following form

$$A_{i,1}h_{i-1} + A_{i,2}h_{i+1} + A_{j,1}h_{j-1} + A_{j,2}h_{j+1} + A_C h_{i,j} = h_{old} + (\Delta t\hat{w})_{mod} \quad (18)$$

where indexes are taken from Fig. 2 and

$$A_{i,1} = -\frac{2\dfrac{\gamma}{\rho_\infty g}}{\left(\Delta\xi_{i-1/2} + \Delta\xi_{i+1/2}\right)\Delta\xi_{i-1/2}} \quad (19)$$

$$A_{i,2} = -\frac{2\dfrac{\gamma}{\rho_\infty g}}{\left(\Delta\xi_{i-1/2} + \Delta\xi_{i+1/2}\right)\Delta\xi_{i+1/2}} \quad (20)$$

$$A_{j,1} = -\frac{2\dfrac{\gamma}{\rho_\infty g}}{\left(\Delta\eta_{j-1/2} + \Delta\eta_{j+1/2}\right)\Delta\eta_{j-1/2}} \quad (21)$$

$$A_{j,2} = -\frac{2\frac{\gamma}{\rho_\infty g}}{\left(\Delta\eta_{j-1/2} + \Delta\eta_{j+1/2}\right)\Delta\eta_{j+1/2}} \quad (22)$$

$$A_C = -A_{i,1} - A_{i,2} - A_{j,1} - A_{j,2} - 1 \quad (23)$$

Note that the distance between the grid points are calculated using projected distances to the plane perpendicular to the gravitation. Eq. (18) is solved iteratively by using line-Gauss-Seidel method. Only one iteration sweep is performed per iteration and the old value of the height is used as a first guess. As a boundary condition for Eq. (18), a zero gradient for h are assumed. In order to avoid oscillation, artificial damping is added to the system by modifying the multipliers A in Eq. (18) as

$$A_{mod} = \max(0.003, A) \quad (24)$$

After this A_C is recalculated from Eq. (23).

Boundary condition for the flow solver

Flow solver boundary conditions are directly put to the cell face. In this way the unnecessary damping of the upwind method was avoided. Pressure is set by using Eq. (9). The velocities are extrapolated by using similar first-order method as used for the surface height equation in Eq. (12). A second-order method Eq. (11) could have been used as well.

Also the surface tension is calculated by using a difference operator. However, this term was found to be relatively small and has only a small stabilazing effect. In order to calculate fluxes at the first computational face (one face from the free surface), the central differences are used.

2.4 Grid Deformation

The deformation of the volume grid follows the shape of a base grid which is copied from the initial grid at the beginning of the simulation. The base grid is kept unchanged during the iteration (Fig. 3).

The grid points float along the grid lines of the base grid. The node-point distributions on each grid line which ends to the free surface are copied from the base grid. In the locations, where the wave goes down the cells along that line must be shrunk. Correspondingly, in the locations where the wave goes up, the cells must be stretched. The stretching factor is obtained simply as a ratio of the total length of a grid line after the free surface deformation and the total length of the corresponding line in the initial grid

$$\phi = l_{\text{new}}/l_{\text{initial}} \quad (25)$$

The grid points are redistributed on the initial grid lines by marching along the lines from the starting points and calculating the distances to each point along

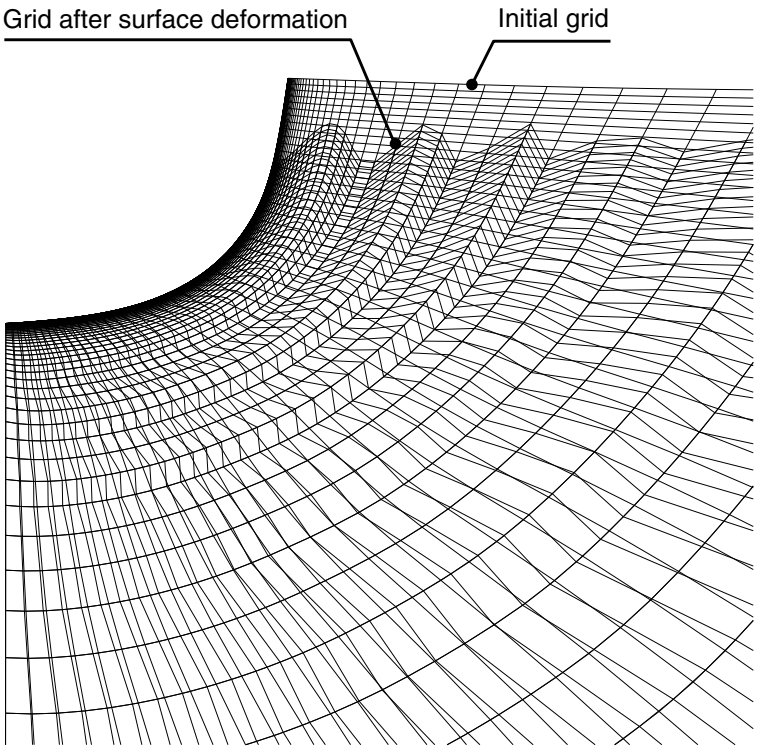

Fig. 3: Grid after a fictive surface deformation.

the polylines. If the wave extends above the initial grid, i.e. the direction of the grid point movements cannot be obtained from the base grid, the grid points are moved in the direction of gravity. The initial grid should be generated high enough in order to avoid such situations.

The grid updating procedure described above requires that the initial grid is carefully designed and generated. The advantages of this method are the speed and robustness. Even the usage of the base grid does not guarantee the positivity of the grid during the deformation. However, the behaviour of the grid points can be better controlled than in the methods in which the volume grid is completely regenerated after each iteration step.

Note that the deformation of the grid does not need to be stretched over the whole block. If the volume grid is constructed from several blocks, the lengths of the overlapping grid lines in neighbouring blocks must be similar, i.e. the blocks can not be partially connected or if they are partially connected, the smaller block defines the part of the bigger block which is modified during the deformation.

2.5 Artificial compressibility method

For the artificial compressibility method Eq. (1) is written in a different form. Let us consider a one dimensional inviscid problem. The momentum equation

can be written in following form by using continuity equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0 \tag{26}$$

The continuity equation can be written as

$$\frac{\partial \rho}{\partial t} + \rho\frac{\partial u}{\partial x} \approx \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0 \tag{27}$$

By using enthalpy and pressure, the continuity equation gets a form

$$\rho_p\frac{\partial p}{\partial t} + \rho_h\frac{\partial h}{\partial t} + \rho\frac{\partial u}{\partial x} + u\left(\rho_p\frac{\partial p}{\partial x} + \rho_h\frac{\partial h}{\partial x}\right) = 0 \tag{28}$$

Also the energy equation can be written using enthalpy and pressure

$$\frac{\partial h}{\partial t} - \frac{1}{\rho}\frac{\partial p}{\partial t} + u\frac{\partial h}{\partial x} - \frac{u}{\rho}\frac{\partial p}{\partial x} \tag{29}$$

By solving the different time derivatives in Eqs. (28) and (29) the following equations are obtained

$$\frac{\partial h}{\partial t} + c_0^2\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x} = 0 \tag{30}$$

$$\frac{\partial p}{\partial t} + \rho c_0^2\frac{\partial u}{\partial x} + u\frac{\partial p}{\partial x} = 0 \tag{31}$$

where

$$c_0^2 = \frac{1}{\frac{\rho_h}{\rho} + \rho_p} \tag{32}$$

Now collect and include v momentum.

$$\frac{\partial V}{\partial t} + A\frac{\partial V}{\partial x} = 0 \tag{33}$$

where $V = (h, u, v, w, p, \phi)^T$ and

$$A = \begin{pmatrix} u & c_0^2 & 0 & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \frac{1}{p} & 0 \\ 0 & 0 & u & 0 & 0 & 0 \\ 0 & 0 & 0 & u & 0 & 0 \\ 0 & \rho c_0^2 & 0 & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & u \end{pmatrix} \tag{34}$$

Variable ϕ is a scalar that is used for differential equation for the turbulence equations.

For the pseudo compressibility, the sound speed is modified as

$$c^2 = (\zeta_t V_{ref}) \tag{35}$$

where V_{ref} represents a reference velocity and ζ_t is a constant. The pseudo sound speed depends on ζ_t and it has an effect to the convergence rate and stability. In these simulations it gets a value of 10. The ρ_p is modified so that the Eqs. (32) and (35) give similar sound speeds.

To change the residuals from conservative to primitive ones and vice versa is not always a trivial task. Here the results are presented

$$\Delta h = \frac{1}{\rho + \frac{\rho_h}{\rho_p}} \left[\Delta E - H\Delta(\rho) + \left(\frac{1}{\rho_p} + u^2 \right) \Delta\rho - u_i \Delta(\rho u_i) \right] \quad (36)$$

$$\Delta u = \frac{1}{\rho} \left[-u\Delta\rho + \Delta(\rho u) \right] \quad (37)$$

$$\Delta v = \frac{1}{\rho} \left[-v\Delta\rho + \Delta(\rho v) \right] \quad (38)$$

$$\Delta p = \frac{1}{\rho_p}\Delta\rho - \frac{\rho_h}{\rho_p}\Delta h \quad (39)$$

$$\Delta\phi = \frac{1}{\rho} \left[-\Delta\rho + \Delta(\rho) \right] \quad (40)$$

where $H = h + \frac{1}{2}u^2 = \frac{E}{\rho} + \frac{p}{\rho}$. The solver writes the equations in a conservative form but it solves the primitive variables T, u, v, w, p and a conservative variable $\rho\phi$. The enthalpy change is converted to the temperature change as

$$\Delta T = \Delta h / c_p \quad (41)$$

$$\Delta(\rho\phi) = \rho\Delta\phi \quad (42)$$

$$(43)$$

where c_p is a specific heat at the constant pressure. The scalar residual is not corrected by the density change because the density change is somewhat corrupted after implicit sweep because of the pseudo-compressibility approach.

2.6 Calculating Cell Face Velocities

In this work, it was found that the original upwinding of fluxes in FINFLO pseudo-compressibility method (not reported) was too diffusive. Waves seemed to loose their shape. A simple new approach for the cell face velocities is suggested. The convective mass fluxes for all equations are calculated as

$$(\rho u)_{i+1/2} = \rho_{i+1/2} u_{i+1/2} - \frac{p_{i+1} - p_i}{\max(c_i, c_{i+1})} \quad (44)$$

where $\rho_{i+1/2}$ is taken as an average of the values in points i and $i+1$. This convective mass flux is applied for all equations. It would be possible to apply this only for the continuity equation, i.e. for themass balance. This was left as a future work. This approach uses the pressure difference as a damping. This

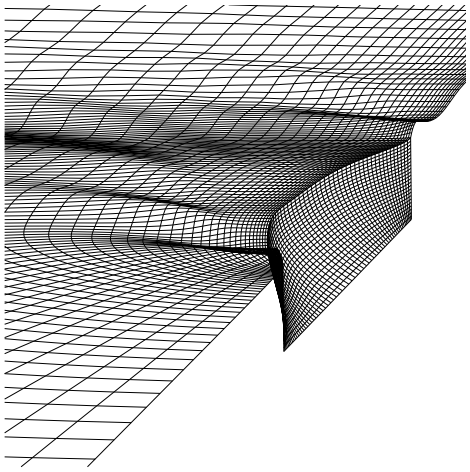

Fig. 4: Surface grid of the Wigley hull and the deformed free surface.

method is somewhat similar that is used in pressure correction methods, but it is a first order in nature, whereas the pressure correction terms are generally third order. This effect also needs some further studies, but is left out of this research.

3 Results

Two test cases were calculated. The First case is the Wigley hull and the second one was the US Navy Combatant DTMB 5415.

3.1 Wigley Hull

This simulation was performed with the Euler equations in order to reduce the number of the grid points. Most of the parameters applied in the surface height equation were designed by using these simulations. The hull was chosen also because there have been previous studies with it [1]. Although the previous calculations were viscid, they give some reference. The hull form is given by the equation

$$y = \frac{B}{2}\left[1 - \left(\frac{2x}{L}\right)^2\right]\left[1 - \left(\frac{z}{T}\right)^2\right] \tag{45}$$

where B, L and T are the breadth, length and draft, respectively. These get values of $L = 4\text{m}$, $B = 0.4\text{m}$ and $T = 0.25\text{m}$.

The grid was of a C-O type having 128 grid points in the flow direction, 64 points perpendicular to the hull and 24 points from the surface. Surface grid of the hull consists of 64×24 points (j-minimum wall). Symmetry of the flow case is used and only half of the ship hull is simulated. The total grid size was $128 \times 64 \times 24 = 196\,608$. Because the Euler computation was used, the first cell size next to the surface could be as large as $y_{n,1} \approx 7\text{mm}$. The figure of the deformed grid at the hull and the surface can be seen in Fig. 4.

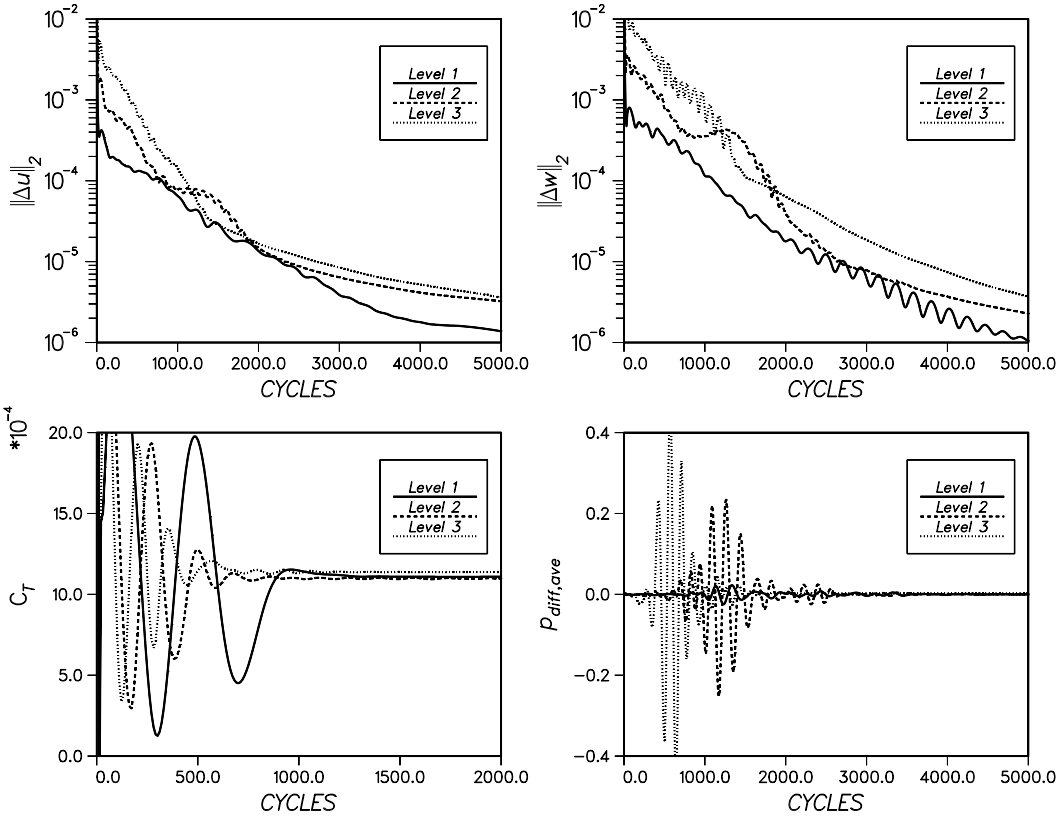

Fig. 5: L_2-norm of the u- and w -velocity residuals (above), and total resistance and averaged pressure (below).

The Froude number was 0.25 ($Fr = u_\infty/\sqrt{gL}$). In order to make comparisons between different grid densities, every other grid line was omited. Hereafter, the densest to coarsest grid simulations are called first, second- and third-level computation. The Courant number was $CFL = 1.5$ for the first level and 2 for the two other grid densities. Four multigrid sweeps were made in the first level simulation and three and two with the coarser ones. The iteration histories of Δu and Δw L_2-norms, the total resistance, and the averaged pressure are seen in Fig. 5. The total resistance is defined as $C_T = F_D / \left(\frac{1}{2}\rho_\infty u_\infty^2 S_{WS}\right)$, where F_D is the drag force and S_{WS} is the wetted surface in the first level computation getting a value of $S_{WS} = 1.18\text{m}^2$ for half model. Note, that this is an inviscid computation, and thus, only a wave drag is simulated. It can be seen that the L_2-norm gets close to the machine accuracy when 5 000 iterations are done. If the total resistance is studied, it can be noted that 1 200 iterations is enough. Some oscillations can be seen in the average pressure. The oscillation comes from the far-field boundary conditions. The computational speed was roughly 65 μs/(cycle·cell) with some unnecessary I/O every iteration cycle. The computation was made on an SGI Origin2000 with 250 MHz R10k.

A most important result of the free surface simulation is the wave profile along the hull. Comparison with different grid densities and also with exper-

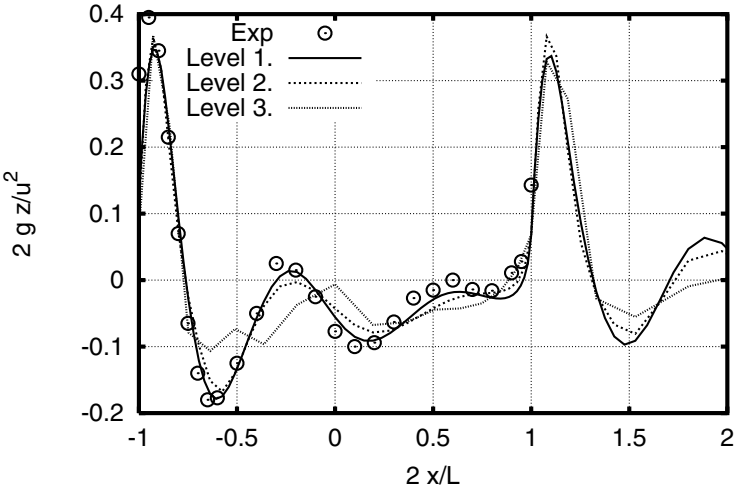

Fig. 6: The computed wave pattern on the Wigley hull. The ship is located between $-1 < 2x/L < 1$.

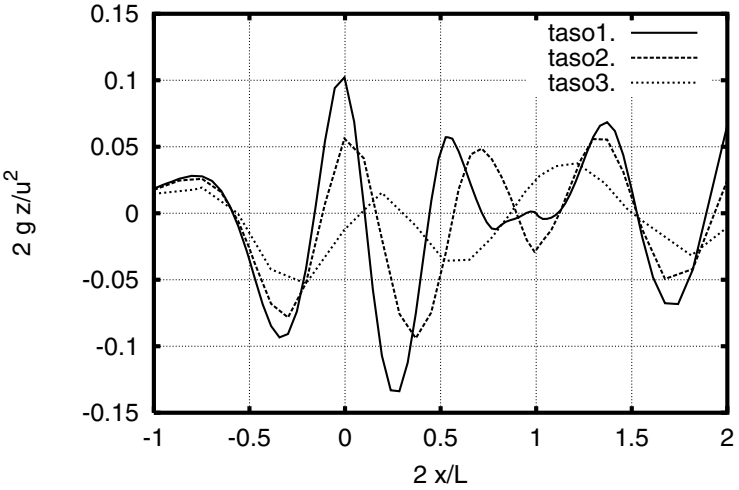

Fig. 7: The computed wave pattern at $y/L = 0.172$ for the Wigley hull.

imental data is shown in Fig. 6. Experimental data is taken from Ref. [1] by using a ruler and pen. Originally experiments where done at the University of Tokyo. Even though the experimental data is naturally taken from a case where viscosity is present, the results are in excellent agreement with the measurements. It is seen that the second-level computation gives good results and only minor improvement comes from the first level computation. The same can be seen in the total resistance (actually only wave resistance in this case) in Fig. 5. If the wave pattern is studied further away from the ship hull, the effect of the grid density can be clearly seen. Figure 7 shows wave cut in location $y/L = 0.172$. As it is seen, the differences are much larger in the wave cut than along the ship hull. The overall wave system is shown in Fig. 8. It

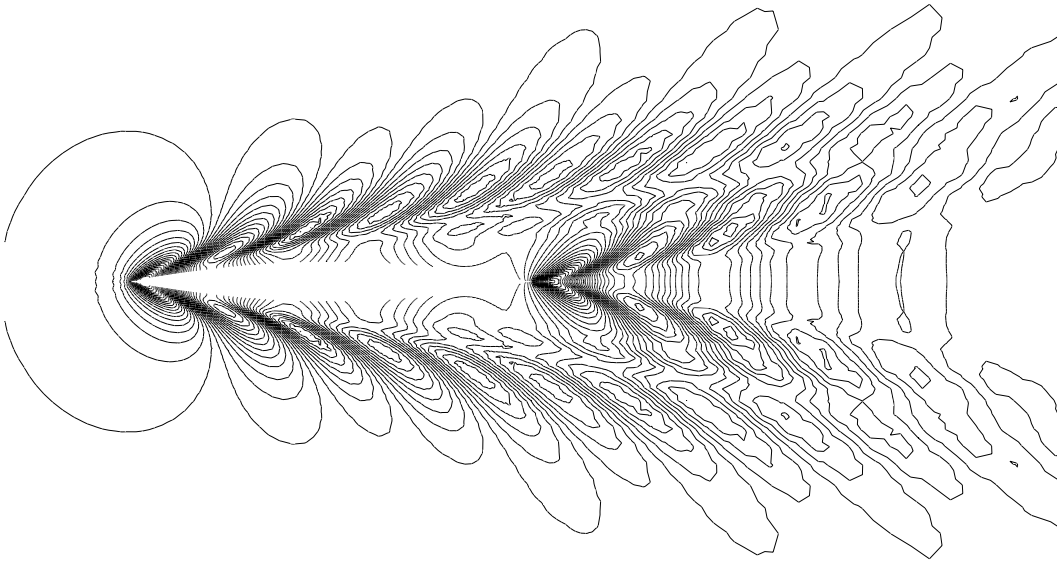

Fig. 8: Computed wave pattern for the Wigley hull.

can be seen that the Kelvin wave pattern is reproduced and results are similar as in Ref. [1].

3.2 US Navy Combatant DTMB 5415

The second hull studied was the US Navy Combatant DTMB 5415. This flow is one of the test cases in the Gothenburg 2000 workshop on 'CFD in Ship Hydrodynamics', where the experimental results are also found (*http://www.iihr.uiowa.edu/gothenburg2000/5415/combatant.html*). The velocity of the hull is $u_\infty = 2.063$m/s, and the length of hull is $L = 5.72$m. The corresponding Froude and Reynolds numbers are $Fr = u_\infty/\sqrt{gL} = 0.28$, and $Re = \rho u_\infty L/\mu_\infty = 12.8 \cdot 10^6$.

The surface and hull grids of the converged solution can be seen in Figs. 9 and 10. The grid is of an O-O type. The grid density is $192 \times 160 \times 48$ ($= 1\,474\,560$) in a flow direction, perpendicular to the hull and perpendicular to the surface, respectively. The first cell height is $3 \cdot 10^{-5}$m. The nondimensional cell height varies from 0.3 to 5 having an averaged value 2.

The Navy Combatant was simulated with three different grid densities. Again the first level is referred as the densest simulation. Figure 11 shows the convergence of different parameters. In this case the residuals do not converge as well as with the simpler Wigley hull. Also the number of iterations needed is somewhat larger, roughly 5 500 iterations, to get a converged total resistance. The computational speed was roughly 70 μs/(cycle·cell) with some unnecessary

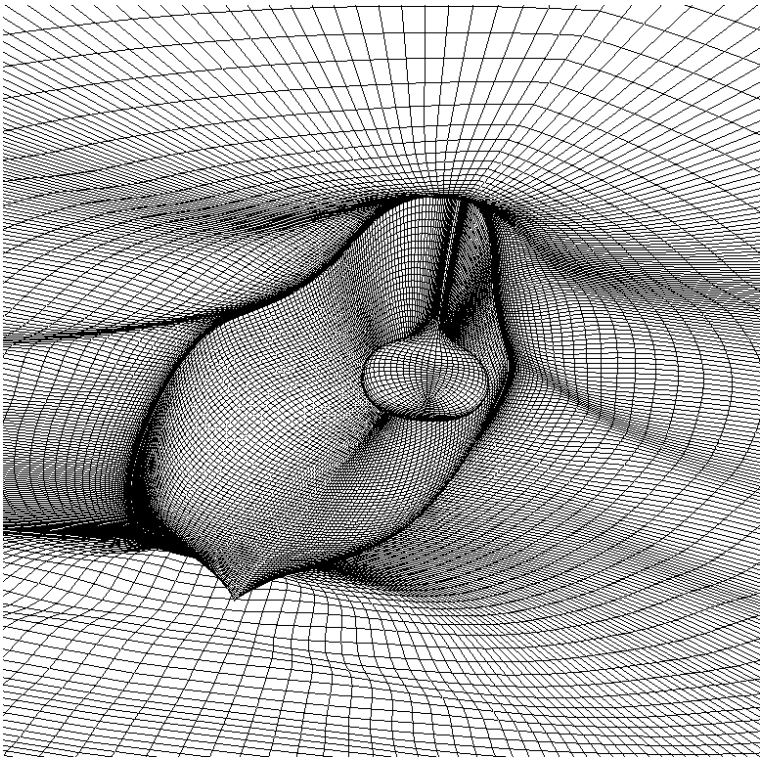

Fig. 9: Surface grid of the US Navy Combatant DTMB 5415.

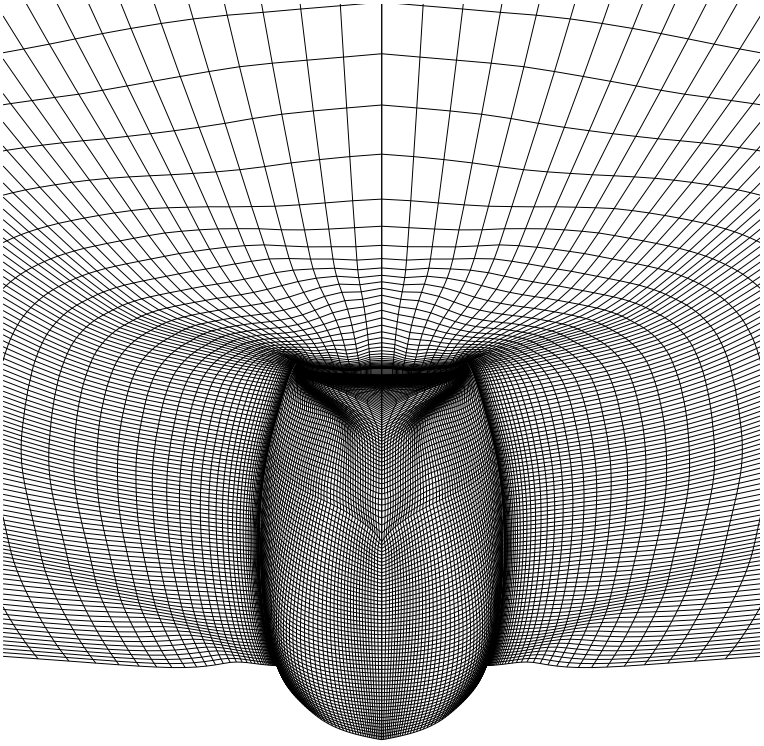

Fig. 10: Surface grid of the US Navy Combatant DTMB 5415.

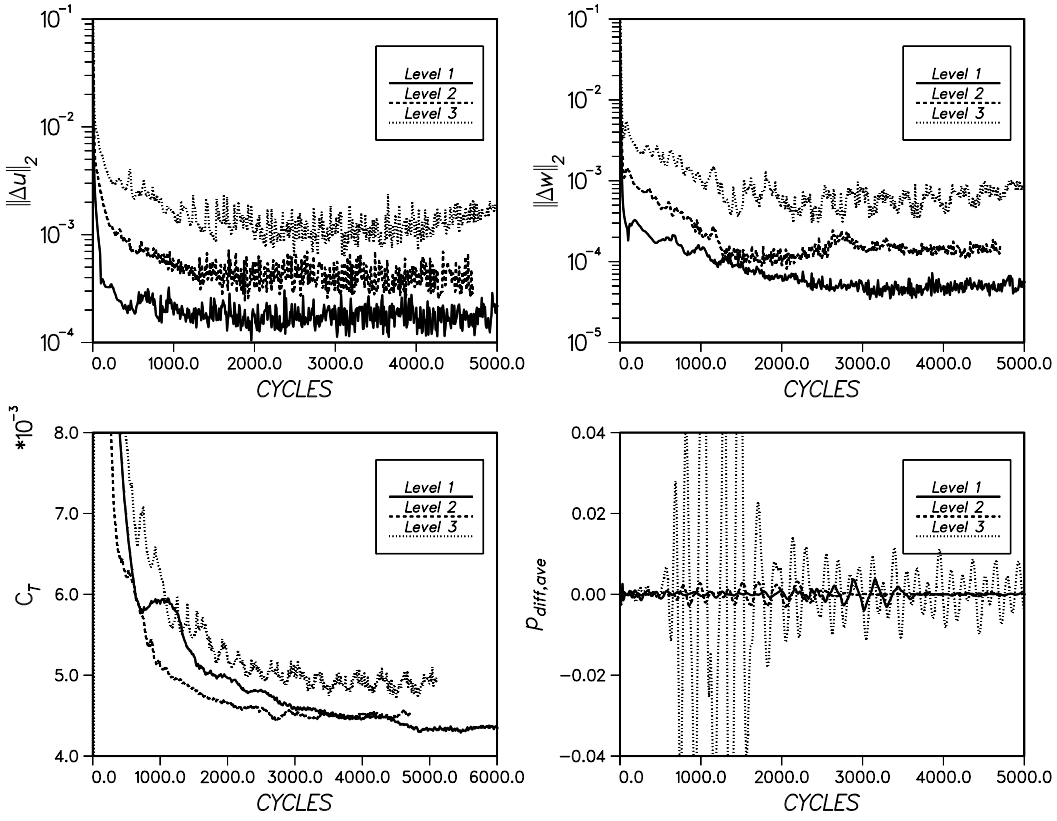

Fig. 11: L_2-norm of the u- and w -velocity residuals (above), and the total resistance and averaged pressure (below) for the US Navy Combatant.

Table. 1: Convergence and force coefficients for the US Navy Combatant.

Grid	CFL	MG	S/L^2	$C_T \cdot 10^3$	$C_f \cdot 10^3$	$C_p \cdot 10^3$	C_L
Exp.			0.1486	4.23	3.31	0.92	0.5188
RefCFD	?	?	0.1440	4.41	3.01	1.40	?
Level 1.	2	3	0.15097	4.321	3.046	1.275	0.524
Level 2.	2	3	0.15070	4.505	3.038	1.467	0.523
Level 3.	2	2	0.14999	4.950	2.868	2.082	0.520

I/O every iteration cycle. The computation was made on an SGI Origin2000 with 250 MHz R10k. The overall computing time needed for the first level simulation was a little bit less than one week. Parallelization would have been needed to reduce the computation period. The average pressure $p_{diff,ave}$ has some oscillations coming from the far field. The convergence and the total force parameters can be seen in Table 1. The force coefficients are made nondimensional by using a reference density, velocity, and the wetted surface area, $S_{DWL} = 4.861\text{m}^2$, from the experiments. Variable S is the wetted surface from the simulation, and the integral variables C_T, C_f, C_p and C_L are the total resistance, the friction resistance, the pressure resistance (wave resistance), and the lift coefficient. The lift coefficient in the experiments is calculated by using

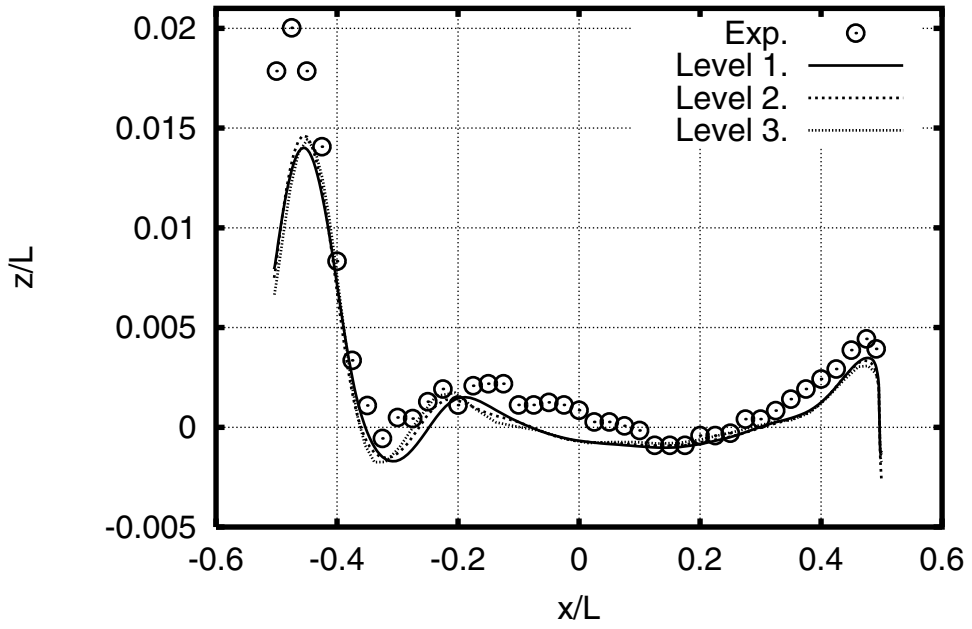

Fig. 12: The computed wave pattern on the Combatant hull.

the knowledge of the model weight. RefCFD computation is taken from the workshop home-pages and it is a CFD result. More details of this case is not known for the authors. The total resistance for the first level computation is 2 % larger than in the experiments. Even though this value is well predicted the difference can be partly explained by the difference in ship floating mass (lift coefficient). If the ship would have let to sink during the simulation, the drag would have been somewhat smaller. Other interesting feature is that the wave resistant C_p is well captured already with the second-level computation. The viscous drag is refined with the first level simulation.

Figure 12 shows the wave profile on a hull surface. This figure shows that the third grid level gives fairly similar results as the first grid level. The shape of the profile is well captured but the level of the bow wave is underestimated. In the experiments, the bow wave brakes which cannot be simulated with the present method. Also the averaged level is somewhat lower than the measured one that might indicate some systematic differences in data handling in experiments and computations. One evidence of some differences in data handling can be seen in the lift coefficient C_L in the Table 1. The smaller coefficient in the experiments indicates that the averaged wave profile should be lower in the experiments. If the wave pattern is studied further away from the ship hull, the effect of the grid density can be clearly seen. Figure 13 shows wave cut in location $y/L = 0.172$. Again the present approach does not predict the level of the sharpest crests and troughs but the improvement can be seen when the grid is refined. In this case the artificial surface tension might damp some of the crest away but it can also be that the grid is not fine enought

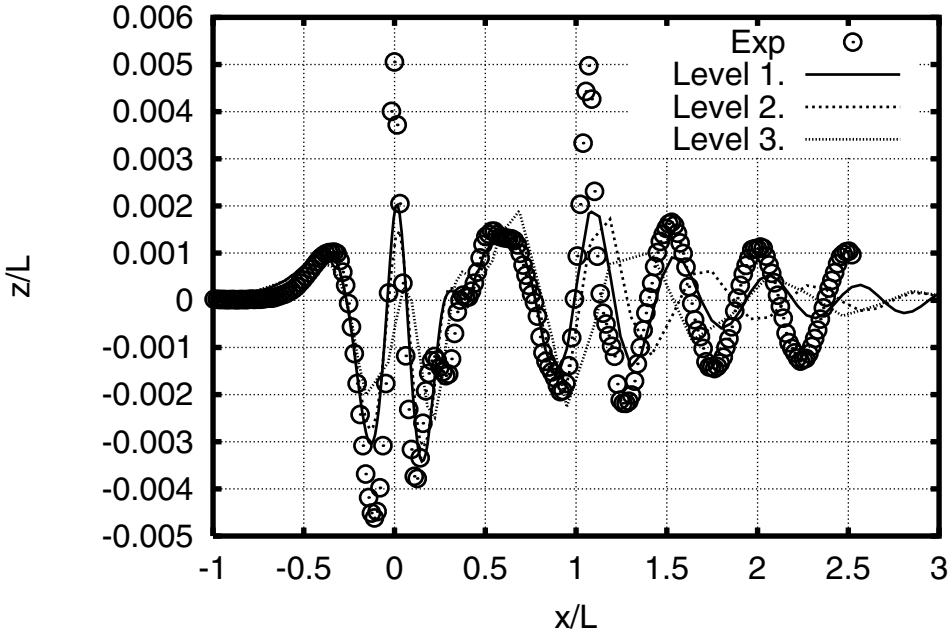

Fig. 13: The computed wave pattern at $y/L = 0.172$ for Combatant hull. The ship is located between $-0.5 < x/L < 0.5$.

to solve the wave pattern accurately. Other explanation could be that in the experiments the bow wave brakes, that cannot be simulated with the surface tracking method. This might change the physics of the flow problem. At the wake the amplitude and the frequency is improved when the finer grids are used.

The overall wave pattern is shown in Fig. 14. Similar findings can be done as with the wave cut (Fig. 13). In the first-level solution some undesirable oscillation takes place and part of it can be seen in the upper portion of the level 1 figure. More detailed comparison can be done by studying separately bow and stern regions in Figs. 15 and 16. Figure 15 depicts that the method predicts a right shape of the bow wave, but the magnitude is somewhat too small. In the experiments the bow wave brakes. The wave shape at the stern area is captured very well. The shape and the height of the stern wave is excellent agreement with experimental data. The velocity magnitude can be studied at the propeller plane at $x/L = 0.435$ at the Fig. 17. It can be seen that the shape of the boundary layer is very similar in the experiments and in the simulation. Tangential velocity vectors at the propeller plane are shown in Fig. 18. Figure 19 depicts a detailed velocity magnitude comparison. Computation and experimental data show a good agreement even though at some locations small differences can be seen. These differences are expected to get smaller if more advanced turbulence modelling would have been used.

Figure 20 shows overall wave pattern for the Combatant. Ship's upper parts are imaginary.

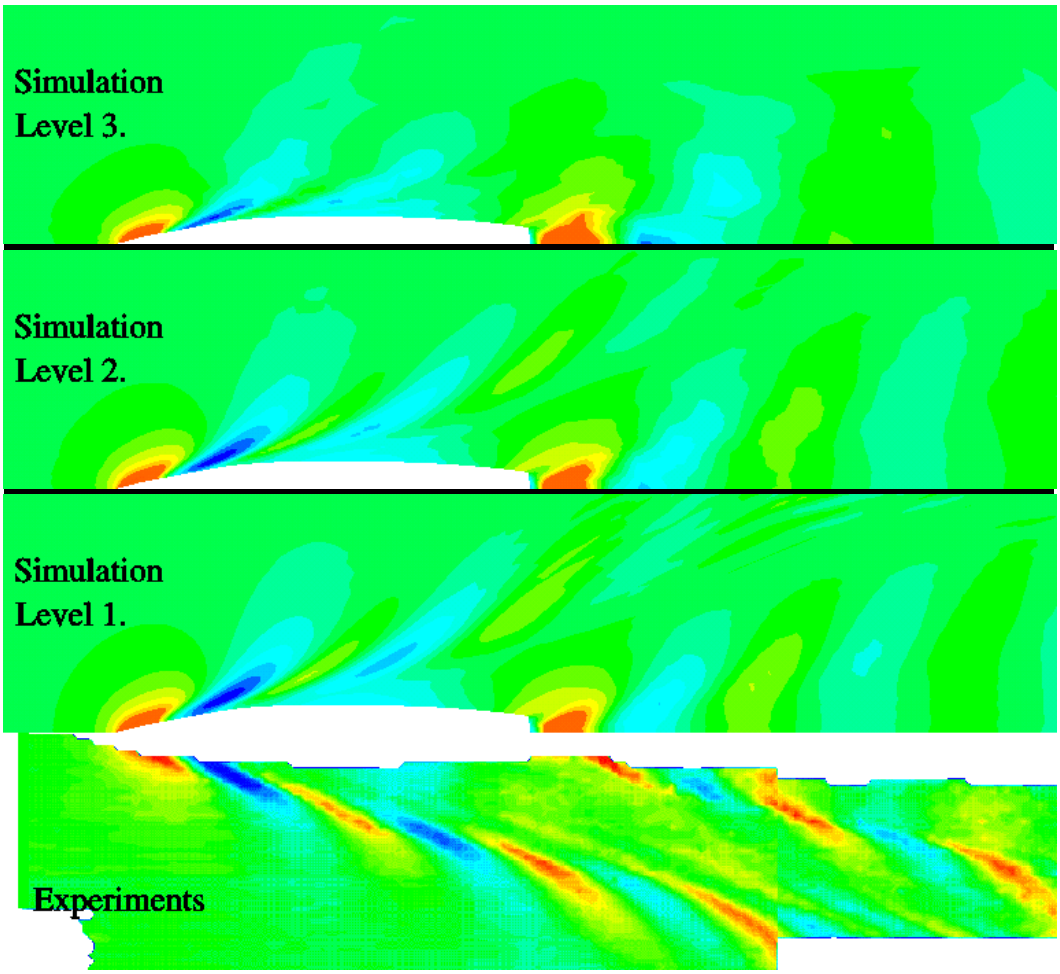

Fig. 14: Computed wave pattern for Navy Combatant with different grid densities and in the experiment.

4 Future Work

Here, a list of future improvements are stated:

- Put the surface height equation into a delta form
- Use a corrected surface velocity of Eq. (44) only for the mass balance
- Far-field boundary conditions for surface height h and/or pressure, could improve convergence
- Parallelization by multiblock strategy
- Change a position of the ship by forces coming from a flow solution
- Modify a COMPUT-file to work with the current problem
- Include back-up grid files similarly as the RSTART-file

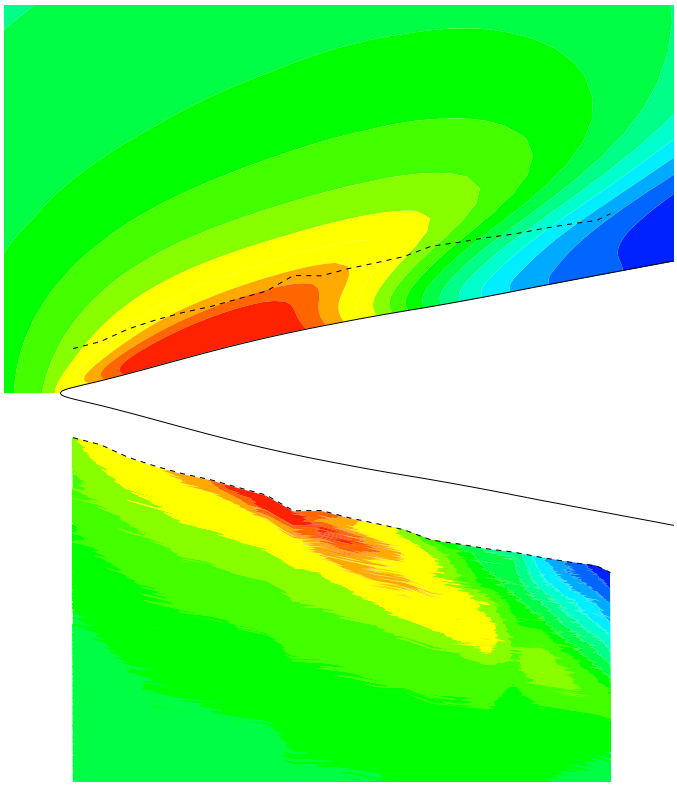

Fig. 15: Wave pattern for Navy Combatant at the bow area. The upper part is from the computation and the lower experiments

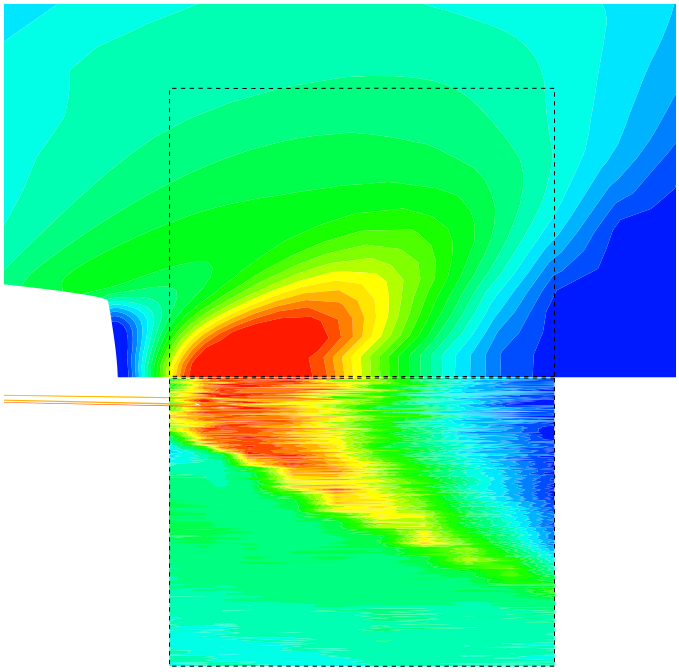

Fig. 16: Wave pattern for Navy Combatant at the stern area. The upper part is from the computation and the lower experiments.

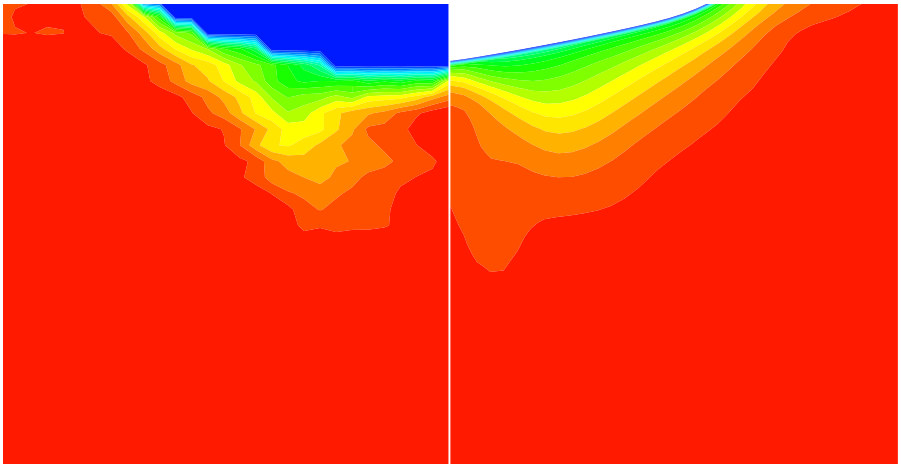

Fig. 17: Velocity magnitude at the propeller plane $x/L = 0.435$. The left-hand side is the experimental data and the right-hand side is the simulated result.

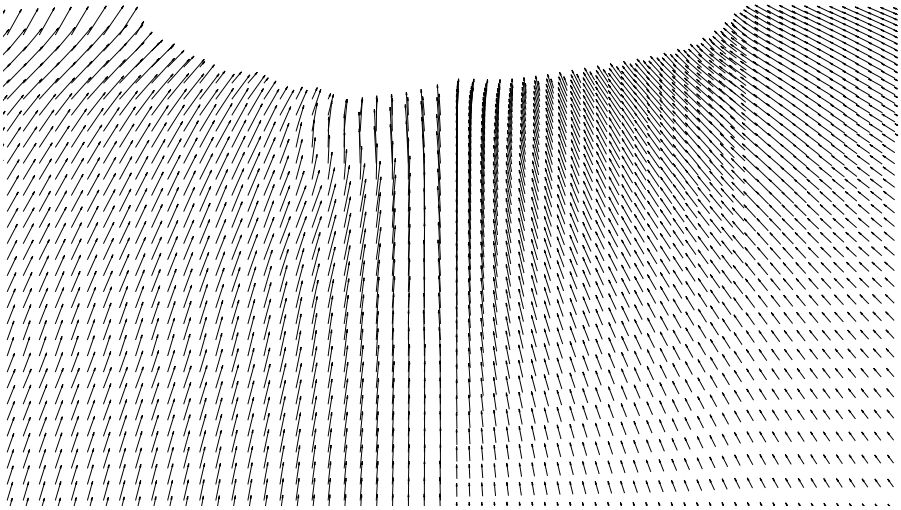

Fig. 18: Tangential velocity vectors at the propeller plane $x/L = 0.435$. The left-hand side is the experimental data and the right-hand side is the simulationed result.

- Make tests with the two-equation turbulence models
- Check the iteration of the LGS-scheme

5 Discussion

Preliminary simulations with a novel free-surface method were presented. Although many open questions are still open, the results are promising as far as the accuracy and convergence rate are concerned.

A surface tracking method was implemented in the flow solver. The method can be divided into three parts: surface tracking, boundary conditions for the

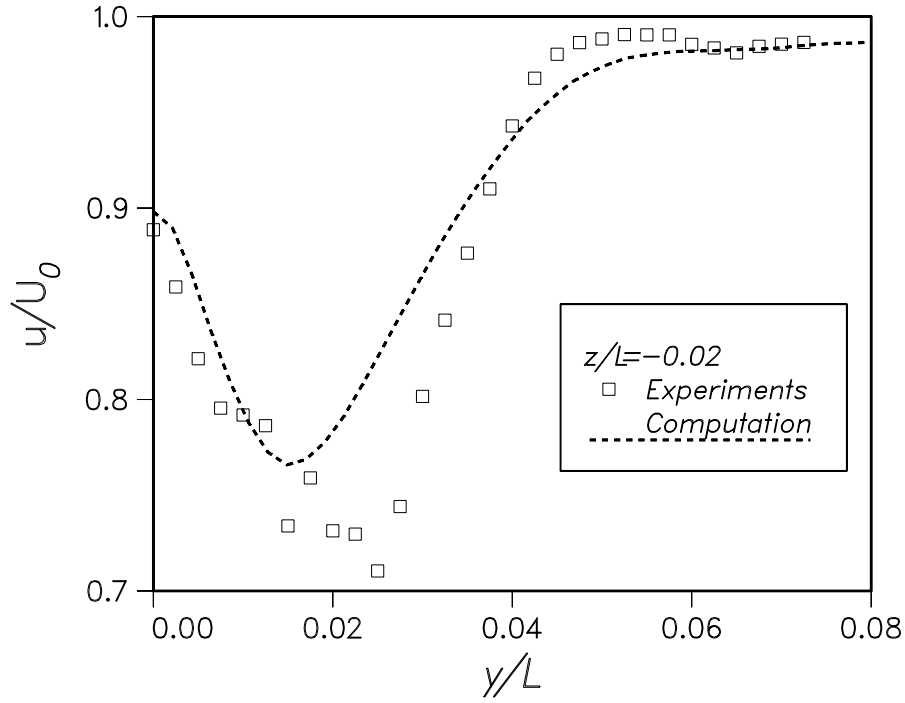

Fig. 19: Axial velocity magnitude at the the propeller plane $x/L = 0.435$ at $z/L = -0.02$.

Fig. 20: Overall wave pattern for the Combatant.

flow solver, and the moving grid technique. The surface tracking is based on a method that takes into account the surface tension in an artificial way, so that the surface height is continuous. Boundary conditions must be given in a proper way, so that they are in balance with the flow solver. Careful tests were done in order to obtain those conditions. The moving grid technique is based on simply moving the grid points along the original grid lines.

The methods does not predict the bow wave strong enough. This can be seen in both test cases. The possible reason is that the grid is not dense enough, the present surface tracking method is too diffusive or the wave brakes in experiments that cannot be solved in present model. Also some undesired oscillation takes place far away from the hull. That probably does not change the results but it surely decreases stability and convergence.

References

[1] Sundel, T., "Computation of the Free Surface Flow Around a Ship Using the NS Solver FINFLO," VTT Manufacturing Technology, 1997.

[2] Siikonen, T., "An Application of Roe's Flux-Difference Splitting for the $k-\epsilon$ Turbulence Model," *International Journal for Numerical Methods in Fluids*, Vol. 21, 1995, pp. 1017–1039.

[3] Rautaheimo, P., Salminen, E., and Siikonen, T., "Parallelization of a Multi-Block Navier–Stokes Solver," in *Proceedings of the Third ECCOMAS Conference*, (Paris), John Wiley & Sons, Ltd., Sept. 1996.

[4] Siikonen, T., Kaurinkoski, P., and Laine, S., "Transonic Flow over a Delta Wing Using a $k-\epsilon$ Turbulence Model," in *Proceedings of the 19th ICAS Congress*, (Anaheim), pp. 700–710, Sept. 1994. ICAS Paper 94-2.3.2.

[5] Rautaheimo, P. and Siikonen, T., "Improved Solid-Wall Boundary Treatment in Low-Reynolds Number Turbulence Models," in *38th AIAA Aerospace Sciences Meeting and Exhibit*, (Nevada, Reno), Jan 2000.

[6] Siikonen, T. and Pan, H., "Application of Roe's Method for the Simulation of Viscous Flow in Turbomachinery," in *Proceedings of the First European Computational Fluid Dynamics Conference*, (Brussels), pp. 635–641, Elsevier Science Publishers B.V., Sept. 1992.

[7] Rautaheimo, P. and Siikonen, T., "Simulation of a Flow in a Return Channel with a Deswirl Cascade Using Advanced Turbulence Modelling," in *Proceedings of the Fourth ECCOMAS Conference*, (Athens), pp. 445–450, John Wiley & Sons, Ltd., Sept. 1998.

[8] Rautaheimo, P., Salminen, E., and Siikonen, T., "Numerical Simulation of the Flow in the NASA Low-Speed Centrifugal Compressor," Helsinki

University of Technology, Laboratory of Applied Thermodynamics, 1998. ISBN 951–22–4360–1.

[9] Rautaheimo, P., Salminen, E., Siikonen, T., and Hyvärinen, J., "Turbulent Mixing Between VVER-440 Fuel Bundle Subchannels: A CFD Study," in *Proceedings of the Ninth International Topical Meeting on Nuclear Reactor Thermal Hydraulics*, (San Francisco, California), American Nuclear Society Inc., Oct. 1999.

[10] Rautaheimo, P., Kaurinkoski, P., and Siikonen, T., "Performance of a Parallel CFD-Code on a Linux Cluster," *CSC News*, Vol. 11, No. 4, 1999, pp. 19–22.

[11] Kaurinkoski, P., Rautaheimo, P., Siikonen, T., and Koski, K., "Performance of a Parallel CFD-Code on a Linux Cluster," in *Proceedings of the Parallel CFD 2000 Meeting*, 2000.

[12] Jameson, A. and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes," *AIAA Journal*, Vol. 24, No. 11, 1986.

[13] Siikonen, T., Hoffren, J., and Laine, S., "A Multigrid *LU* Factorization Scheme for the Thin-Layer Navier–Stokes Equations," in *Proceedings of the 17th ICAS Congress*, (Stockholm), pp. 2023–2034, Sept. 1990. ICAS Paper 90-6.10.3.

[14] Roe, P., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[15] Van Leer, B., "Flux-Vector Splitting for the Euler Equations," in *Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics*, (Aachen), 1982. (also Lecture Notes in Physics, Vol. 170, 1982).

[16] White, F. M., *Viscous Fluid Flow*. New York: McGraw–Hill Book Co., 2nd ed., 1991. ISBN 0–07–069712–4.